Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №17

Указатели и ссылки при работе с функциями

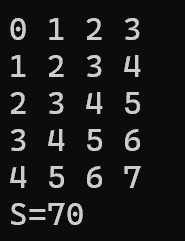
Выполнил: Мышковец Артём Витальевич 10ПИ

Минск 2024

Задание 1.

Выполнив программу, записанную в правой части, изучить один из способов передачи *динамической* *матрицы* в функцию пользователя.

Записать условие задачи.

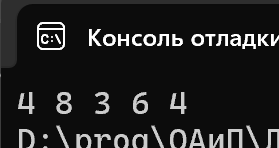


Условие:

Посчитать сумму всех элементов матрицы

Задание 2.

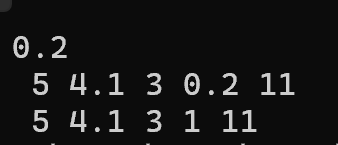
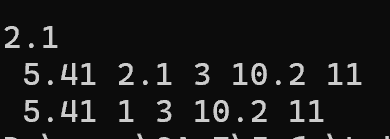
Выполнив программу, записанную в правой части, изучить один из способов передачи *одномерного* *массива* в функцию, а также использование *указателя как результата* выполнения функции.



Задание 3.

Изучить использование *ссылки* как *результата* работы функции, выполнив программу, записанную в правой части.

Опробовать работу программы с разными значениями массива **А**.

Задание 5.

В соответствии со своим вариантом написать главную функцию, в которой имеются вызовы ***функций пользователя***, реализующих задачи из таблицы ниже. Ввод исходных данных и вывод результатов осуществить в главной функции, при этом использовать***динамические* *массивы***.

Для передачи параметров в функции пользователя и возвращения результатов применить ***указатели*** и ***ссылки***.

**Вариант 13**

1. Дан массив **A** вещественного типа, содержащий 20 положительных и отрицательных элементов. Сформировать массив **B** из положительных элементов массива **A**, имеющих четный индекс. Найти сумму квадратов элементов нового массива.

2. Дана целочисленная квадратная матрица. Если минимальное значение находится на главной диагонали, то вывести его значение и индексы.

#include <iostream>

using namespace std;

int& task1(int A[]) {

int\* B = new int[10];

for (int i = 0,j=0;i <= 20;i += 2) {

if (A[i] > 0) {

B[j++] = A[i];

}

}

int sum = 0;

for (int i = 0;i != 10;++i) {

sum += B[i] \* B[i];

}

return sum;

}

int\* getMin(int\* B[], int size) {

int\* minPtr = B[0];

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

if (B[i][j] < \*minPtr) {

minPtr = B[i] + j;

}

}

}

return minPtr;

}

int main() {

setlocale(LC\_ALL, "RU");

cout << "Выберите задание(1,2): ";

int choice;

int res=0;

int size=0;

int\* A;

cin >> choice;

switch (choice)

{

case 1:

A = new int[20];

for (int i = 0;i != 20;++i) {

cout << "A[" << i << "]: ";

cin >> A[i];

}

res = task1(A);

cout << "Сумма квадратов: " << res;

delete[] A;

break;

case 2:

int\*\* B;

cout << "Введите размер: ";

cin >> size;

B = new int\* [size];

for (int i = 0; i != size; i++)

{

B[i] = new int[size];

}

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

cout << "B[" << i << "][" << j << "]: ";

cin >> B[i][j];

}

}

int\* min = getMin(B, size);

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

if ((B[i] + j) == min) {

cout << "Индексы: " << i << " " << j << endl << "Значение: " << \*min;

}

}

}

for (int i = 0; i != size; i++)

{

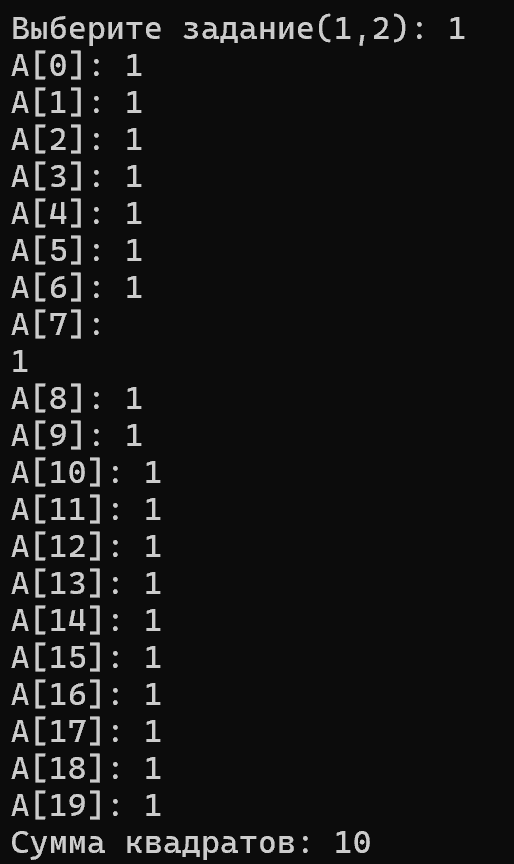
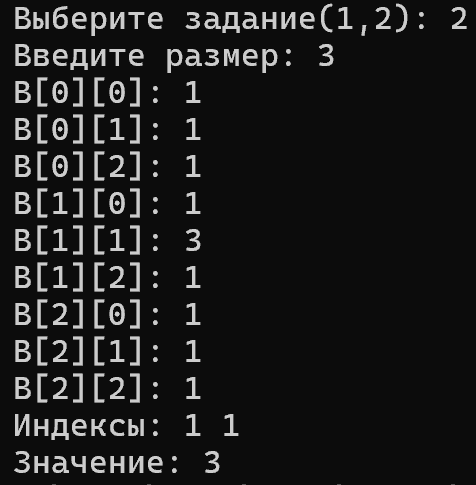
delete[] B[i];

}

delete[] B;

}

}

**Дополнительные задания**

**Вариант 6**

1. Найти сумму минимального и максимального элементов одномерного массива.

2. Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые.

//var6

#include <iostream>

using namespace std;

int& task1(int A[]) {

int min = A[0];

for (int i = 0;i != 15;++i) {

if (A[i] < min) {

min = A[i];

}

}

int max = A[0];

for (int i = 0;i != 15;++i) {

if (A[i] > max) {

max = A[i];

}

}

int sum = max + min;

return sum;

}

bool task2(int\* B[], int size) {

for (int i = 0;i != size;++i) {

int counter = 0;

for (int j = 0;j != size;++j) {

if (B[i][j] == 0) {

++counter;

}

}

if (counter == 0) {

return false;

}

}

return true;

}

int main() {

setlocale(LC\_ALL, "RU");

cout << "Выберите задание(1,2): ";

int choice;

int res = 0;

int size = 0;

int\* A;

cin >> choice;

switch (choice)

{

case 1:

A = new int[15];

for (int i = 0;i != 15;++i) {

A[i] = rand() % 200;

}

res = task1(A);

cout << "Сумма: " << res;

delete[] A;

break;

case 2:

int\*\* B;

cout << "Введите размер: ";

cin >> size;

B = new int\* [size];

for (int i = 0; i != size; i++)

{

B[i] = new int[size];

}

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

cout << "B[" << i << "][" << j << "]: ";

cin >> B[i][j];

}

}

bool isTrue = task2(B, size);

if (!isTrue) {

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

if (B[i][j] < 0) {

B[i][j] = 0;

}

}

}

}

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

cout << B[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i != size; i++)

{

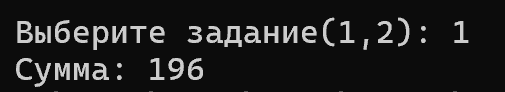
delete[] B[i];

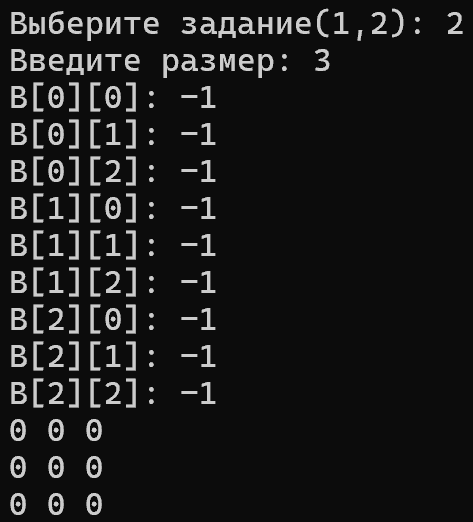
}

delete[] B;

}

}

  
(массив вводится автоматически псевдослучайными числами от 1 до 200)



**Вариант 11**

1. Изменить одномерный массив, вычеркнув из него нечетные элементы.

2. Дана целочисленная квадратная матрица. Если она не содержат отрицательных элементов, то определить сумму элементов в тех строках, где отрицательные элементы отсутствуют.

//var11

#include <iostream>

using namespace std;

void deleteElement(int A[], int pos, int endArr) {

for (int i = pos;i < endArr - 1;++i) {

A[i] = A[i + 1];

}

}

int& task1(int A[]) {

int len = 15;

for (int i = 0;i < len;++i) {

if (A[i] % 2 != 0) {

deleteElement(A, i, len);

--i;

--len;

}

}

return len;

}

int& task2(int\* B[], int size) {

int sum = 0;

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

sum += B[i][j];

}

}

return sum;

}

int main() {

setlocale(LC\_ALL, "RU");

cout << "Выберите задание(1,2): ";

int choice;

int reslen = 0;

int size = 0;

int\* A;

cin >> choice;

switch (choice)

{

case 1:

A = new int[15];

for (int i = 0;i != 15;++i) {

A[i] = rand() % 200;

}

cout << "Массив до изменений: " << endl;

for (int i = 0;i != 15;++i) {

cout << A[i] << "(" << i+1 << ") ";

}

cout << endl;

reslen = task1(A);

cout << "Массив после изменений:" << endl;

for (int i = 0;i != reslen;++i) {

cout << A[i] << "(" << i+1 << ") ";

}

delete[] A;

break;

case 2:

int\*\* B;

cout << "Введите размер: ";

cin >> size;

B = new int\* [size];

for (int i = 0; i != size; i++)

{

B[i] = new int[size];

}

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

cout << "B[" << i << "][" << j << "]: ";

cin >> B[i][j];

}

}

int ressum = task2(B, size);

cout << "Сумма: " << ressum;

for (int i = 0; i != size; i++)

{

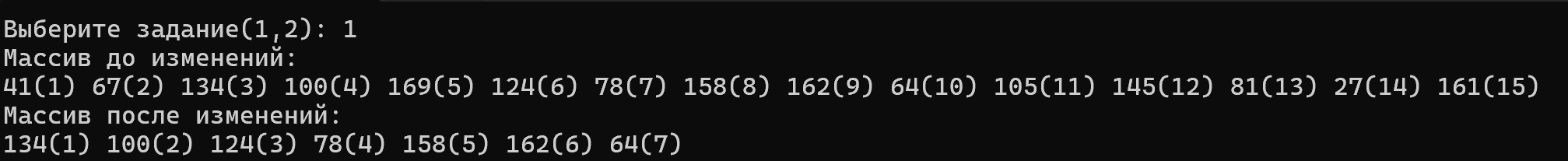
delete[] B[i];

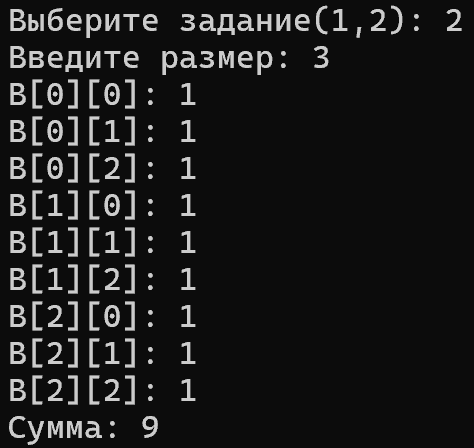
}

delete[] B;

}

}





**Вариант 4**

1. Создать одномерный массив, содержащий 15 элементов, наполнить его случайными значениями в интервале от 1 до 200. Определить сумму всех нечетных элементов массива.

2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.

//var4

#include <iostream>

using namespace std;

int& task1(int A[]) {

int sum = 0;

for (int i = 0;i != 15;++i) {

sum += A[i];

}

return sum;

}

int& getPos(int\* B[], int size) {

int row = 0;

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

if (B[i][j] > 0) {

row = i;

}

}

}

return row;

}

int main() {

setlocale(LC\_ALL, "RU");

cout << "Выберите задание(1,2): ";

int choice;

int res = 0;

int size = 0;

int\* A;

cin >> choice;

switch (choice)

{

case 1:

A = new int[15];

for (int i = 0;i != 15;++i) {

A[i] = rand() % 200;

}

res = task1(A);

cout << "Сумма: " << res;

delete[] A;

break;

case 2:

int\*\* B;

cout << "Введите размер: ";

cin >> size;

B = new int\* [size];

for (int i = 0; i != size; i++)

{

B[i] = new int[size];

}

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

cout << "B[" << i << "][" << j << "]: ";

cin >> B[i][j];

}

}

int row = getPos(B, size);

int pos = row == 0 ? size - 1 : row - 1;

for (int i = 0;i != size;++i) {

B[pos][i] \*= -1;

}

cout << "Индекс строки: " << row << endl;

for (int i = 0;i != size;++i) {

for (int j = 0;j != size;++j) {

cout << B[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i != size; i++)

{

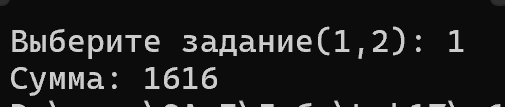
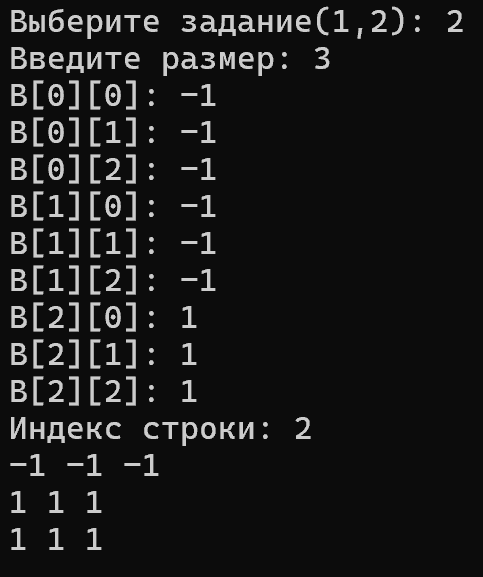
delete[] B[i];

}

delete[] B;

}

}

**** ****